RCA PF020154 Ref AC

CITE[illegible] APPLICANT

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
11 October 2001 (11.10.2001)

PCT

(10) International Publication Number
WO 01/75634 A1

(51) International Patent Classification[7]: G06F 15/173, 15/16

(21) International Application Number: PCT/US01/09780

(22) International Filing Date: 26 March 2001 (26.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/541,313 1 April 2000 (01.04.2000) US

(71) Applicant: TELLUS TECHNOLOGY, INC. [US/US]; 6140 Stevenson Boulevard, Fremont, CA 94538 (US).

(72) Inventors: SLOAN, Mark, Kent; 1277 Stanhope Lane, Hayward, CA 94545 (US). PON, Rayman, W.; 1451 Poppy Way, Cupertino, CA 95014 (US).

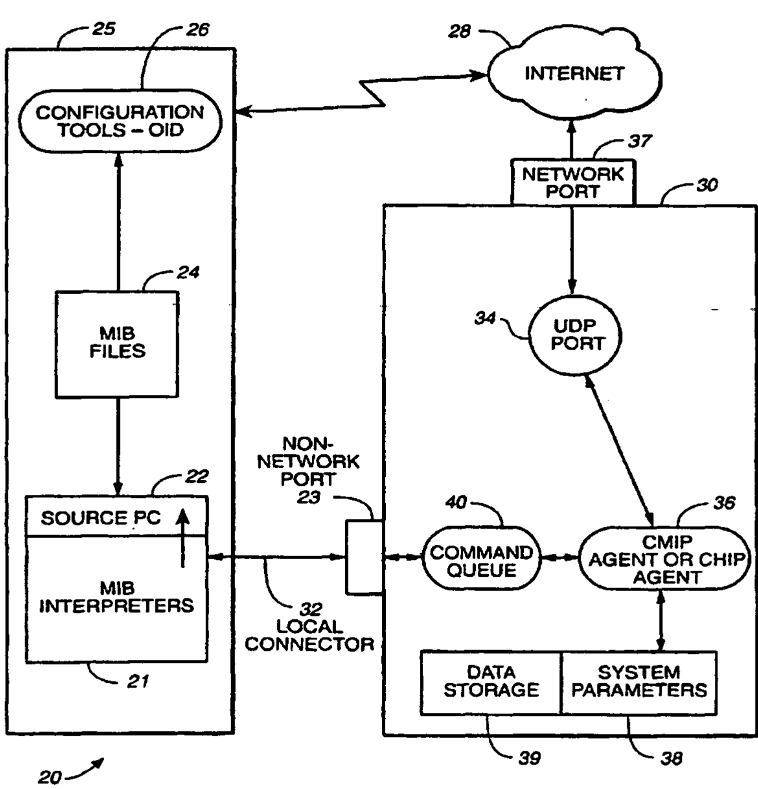
(74) Agent: TANKHILEVICH, Boris, G.; Law Offices of Boris G. Tankhilevich, 536 North Civic Drive #A, Walnut Creek, CA 94596 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: SYSTEM UTILIZING A NETWORK MANAGEMENT PROTOCOL FOR DUAL LOCAL/REMOTE NETWORK ACCESS



(57) Abstract: A network dual access system (20) and method for accessing a network/dual/access programmable device (30) in a dual local/remote mode by using a network/dual/access programmable device (30). The network/dual/access programmable device (30) in the preferred embodiment comprises a local/remote Simple Network Management Protocol (SNMP) Agent (36) configured to interact with the local/remote/interface device (25) in the dual local/remote access mode utilizing a Simple Network Management Protocol (SNMP). In alternative embodiment of the present invention, the network/dual/access programmable device (30) comprises a local/remote Common Management Information Protocol (CMIP) Agent configured to interact with the local/remote/interface programmable device (25) in the dual local/remote access mode utilizing a Common Management Information Protocol (CMIP).

WO 01/75634 A1

Published:
— *with international search report*

Description

SYSTEM UTILIZING A NETWORK MANAGEMENT PROTOCOL FOR DUAL LOCAL/REMOTE NETWORK ACCESS

FIELD OF THE INVENTION

The current invention is in the field of network management protocols, and more specifically, in the field of the simple network management protocols (SNMP).

BACKGROUND ART

In the available art, there are Internet protocols for setting values for and getting values from Internet-based programmable devices, like a server, or an Internet appliance. Thus, for example, one uses the Internet-based protocol to get the length of time an Internet-based unit was turned on or to set the time length to the next re-set of the Internet-based unit. There are also enterprise dependent variables. For example, in a sprinkler network management system, the example of the enterprise dependent variables are: {the number of sprinklers N that should be turned off or on, or the id of each particular sprinkler I1, I2, ...IK that has to be turned off or on}, wherein N and K are integers.

In the prior art, the SNMP protocol is widely used for getting and setting values from an Internet-based programmable device. The advantage of the prior art SNMP protocol usage was the opportunity for a user to buy a number of configuration tools, including an open view interface (OVI), from a vendor and to write a proprietary management information base (MIB) module (MIB file) that can be plugged in into interface. The OVI interface reads the MIB files in order to obtain the proprietary information about the Internet-based programmable device and in order to use the proprietary information to manage

the Internet-based programmable device. For instance, Hewlett-Packard (HP) sells the open view interface (OVI) for SNMP protocol. Another advantage of using SNMP protocol is the ability to interrogate the Internet-based programmable device by using the SNMP protocol without prior knowledge of the particular programmable device. This is possible because the SNMP protocol uses the *get/next* model of interrogation in its MIB files. In the *get/next* model of interrogation the SNMP protocol uses different MIB files written for different applications but all having the same topological structure as a tree of different variables, but with the same trunk.

However, the prior art method of using the SNMP protocol does not allow a user to interrogate the Internet-based programmable device locally, without Internet access, and using only the physical proximity to the programmable device. Indeed, the service technician has two options: (a) to service the Internet-based programmable device remotely by dialing the Internet access number and using the SNMP protocol, or (b) to service the Internet-base programmable device locally by using a computer, cable, and a service program without using the Internet access, and without using the SNMP protocol. The problem with this approach is that for each new programmable device, or for each new usage of the same programmable device, the service technician has to change the service program in order to access and to service the programmable device locally.

What is needed is to extend the usage of the SNMP protocol so that the service technician can use the same SNMP protocol in a remote mode and/or in a local mode in order to access and to service any new Internet - based programmable device, or the same Internet-based programmable device that has new applications, without re-writing the local service program.

SUMMARY OF THE INVENTION

To address the shortcomings of the available art, the present invention provides a network dual access (local/remote) system that allows the service technician to access and to service any new Internet -based programmable device both locally and remotely.

More specifically, one aspect of the present invention is directed to a network dual access system comprising: (1) a network_dual_access programmable device configured to be accessed using a dual local_remote mode; and (2) a local_remote_interface device coupled to the network_dual_access programmable device; wherein the network_dual_access programmable device utilizes the local_remote_interface device for information transfer in the dual local_remote access mode.

In one embodiment of the present invention, the network_dual_access programmable device comprises a local_remote Simple Network Management Protocol (SNMP) Agent (L_R_SNMP_Agent) configured to interact with the local_remote_interface device in the dual local_remote access mode utilizing a Simple Network Management Protocol (SNMP). In an alternative embodiment of the present invention, the network_dual_access programmable device comprises a local_remote Common Management Information Protocol (CMIP) Agent (L_R_CMIP_Agent) configured to interact with the local_remote_interface programmable device in the dual local_remote access mode utilizing a Common Management Information Protocol (CMIP).

In the preferred embodiment of the present invention, the network_dual_access programmable device further includes: (a) a data conversion management module configured to convert a set of data transferred from and/or to the MIB file using the SNMP protocol; and (b) a data storage system configured to store a set of network_dual_access programmable device-based parameters that are related to a set of local_remote_interface device-based parameters stored in a management information base (MIB) file. In the preferred embodiment, the

data conversion management module further includes an SNMP interpreter configured to interpret a set of data in an Abstract Syntax Notation One (ASN.1) format transferred to and /or from the MIB file.

In one embodiment of the present invention, the local_remote_interface device includes: (a) a management information base (MIB) file including a set of variables configured to be transferred to and/or from the network_dual_access programmable device in the dual local_remote access mode; (b) a configuration management tool configured to read and/or write the set of variables using the management information base (MIB) file in order to configure the graphic user interface (GUI); and (c) a Simple Network Management Protocol (SNMP) Client configured to access the network_dual_access programmable device in the dual local_remote access mode utilizing the Simple Network Management Protocol (SNMP).

In one local mode embodiment of the present invention, the local_remote_interface device includes a local connector configured to exchange a set of data between the MIB and the network_dual_access programmable device using the SNMP protocol and bypassing the Internet. In one remote mode embodiment of the present invention, the local_remote_interface device includes a remote connector configured to exchange a set of data between the MIB and the network_dual_access programmable device using the SNMP protocol and via the Internet.

Another aspect of the present invention is directed to a method for a dual local_remote SNMP access of a network_dual_access programmable device using a local_remote_interface device. In one embodiment, the method comprises the following steps: (1) transferring an Abstract Syntax Notation One (ASN.1) format derived from a management information based (MIB) file including a set of variables from the network_dual_access programmable device to the local_remote_interface device; (2) emulating an SNMP transaction by utilizing the MIB file and by using a Simple Network Management Protocol (SNMP) Client

installed on the local_remote_interface device; (3) transmitting the emulated SNMP transaction from the local_remote_interface device to a data conversion module installed on the network_dual_access programmable device; and (4) converting the emulated SNMP transaction into a set of data readable by an Agent installed on the network_dual_access programmable device.

BRIEF DESCRIPTION OF THE DRAWINGS

The aforementioned advantages of the present invention as well as additional advantages thereof will be more clearly understood hereinafter as a result of a detailed description of a preferred embodiment of the invention when taken in conjunction with the following drawings.

FIG. 1A depicts a prior art local system architecture designed to service a programmable device.

FIG. 1B shows a system of the present invention that allows to service a programmable device in both local and remote modes.

FIG. 1C depicts a prior art SNMP model of a managed network.

FIG. 2A shows a prior art diagram illustrating the retrieval and modification operations utilizing a *request* message and a *response* message.

FIG. 2B is a prior art diagram depicting the first event report operation using a single message *trap*.

FIG. 2C depicts a prior art diagram illustrating the second event report operation using the *inform* message and the *response* message.

FIG. 3 illustrates the prior art rules for writing MIB modules that come from two sources: (1) ASN.1 modules defining macros and base type specifications; and (2) textual description that modify the ASN.1 modules.

FIG. 4 is a flow chart of a method of the present invention for the entire SNMP client-server dual mode access model.

FIG. 5 illustrates a flow chart of a method of the present invention that is performed by the SNMP Client side of the SNMP client-server dual mode access model.

BEST MODE FOR CARRYING OUT THE INVENTION

FIG. 1A depicts a prior art local system architecture 10 designed to service a programmable device 16. The technician is located in physical proximity to the programmable device 16 and has a local access to the programmable device 16. In order to service the programmable device 16, the technician uploads the programmable device parameters 18 to his PC/laptop 12, writes or uses the already written proprietary service program 11, and downloads the program 11 to the command interpreter 14 that interprets the program commands to the programmable device 16. The problem with this approach is that if additional parameters are to be installed on the programmable device 16, the technician has to change the service program 11.

In the preferred embodiment of the present invention, FIG. 1B shows the system 20 that allows to service the programmable device 30 in both local and remote modes. This is done by extending the usage of the Simple Network Management Protocol (SNMP) that was primary used in the prior art for remote (including Internet-based) applications, to the local usage in order to access and to service any programmable device in both remote and local mode. The dual SNMP local/remote access system (and method) presented in the current patent application allows a user to stop using a local service program that has to be adjusted every time when a new feature or parameter are added to the programmable device 30. Instead, the user equipped with a dual SNMP local/remote access system 20 of the present invention

can access the programmable device 30 via the Internet 28, or locally via the serial connector 32, but without having to re-write the service program. The job of finding out the new system parameters 38 is done by using the SNMP interrogation tools. (See detailed discussion below). After the new state of the programmable device 30 is determined, the programmable device can be serviced both locally and/or remotely (via Internet) by using the same universal SNMP protocol and SNMP Agent 36. In an alternative to the dual access embodiment, a custom application program interface, such as API, has to be defined, tested, and maintained. An implementation of such an API would require additional resources, like RAM and ROM (read only memory).

FIG. 1C depicts a prior art model 50 of an SNMP-based managed network that includes a number of elements. Each of managed nodes 54, 58, 60 includes a processing entity called an Agent 70. Each management station 52 includes at least one processing entity called management application 72 (or manager). In an alternative embodiment, a management station includes a dual-role entity 56 that is able to perform in both manager and Agent roles. Management information 74 in each managed nodes 54, 58, 60 describes the configuration, state, and statistics that control the actions of a managed node. A management protocol is utilized by the managers and Agents to exchange messages during management communication sessions 62, 64, or 66. This prior art model of SNMP-based management is applied to the system 20 of the present patent application in order to service the programmable device 30 in both remote mode via Internet, and in a local mode bypassing Internet altogether.

The SNMP-based management model is defined by a collection of documents. These documents define a management framework including four major components: (1) a management protocol; (2) a definition of management information and events; (3) a core set of management information and events; and (4) a mechanism and approach to

manage the use of the protocol including security and access control.

The management protocol defines the format and meaning of the management communications between the SNMP processing entities. There are two major versions of the SNMP protocol. The first version is called SNMPv1, the second is called SNMPv2, and the third is called SNMPv3. The operations in SNMP are limited to retrieving the value of management information, modifying the value of management information, and reporting an event. Each class (or type) of management information is assigned a unique identity. An instance from a class of management information is called a variable. Each variable is given a unique identity based on the identity of its class and its identification within its class. Each class (or type) of event is assigned a unique identity.

There is one modification operation which is called a *set*. The operand to a *set* is a list of pairs. Each pair includes the identity of a variable and its desired value. *Set* operations are used to configure and control a managed system. There are two types of retrieval operations. Both retrieve the value of variables. For both the result is a list of pairs, wherein each pair includes the identity of a variable and its current value. Both types have as operands a list of identities. The first type of retrieval operation requires that the identities are those that match exactly the identity of returned variables. This retrieval operation is used when the identity for each variable is known. There is a single operation of this type, called *get*. The second type of retrieval operation uses each identity in a request as an approximation of the identity for a variable. Each returned identity is the one assigned to the first accessible variable whose identity is greater than the given identity. There are two operations of this type, called *getnext* and *getbulk*. These operations are used when the identity for a class of management information is known, but the identity for each

accessible instance within that class is not known. These operations may be also used to determine each class of management information including accessible instances. The *get, getnext* and *getbulk* operations are used to monitor a managed system. There are two event-reporting operations, called *trap* and *inform*. Each of these operations specify an event and a list of pairs. A pair includes the identity of a variable and its value. These operations are used to report the occurrence of events on a managed system to a list of managers configured to receive events for that managed system.

SNMP *operations* occur through *message exchange* over a message transport service. In one embodiment, the format of messages is defined using a subset of the Abstract Syntax Notation One (ASN.1) language. This format, called an abstract syntax, is independent of the representation of data on any particular system. To transmit a message, it should be first converted into a string of octets (bytes). A transfer syntax specifies the format of converted data. SNMP uses a subset of the Basic Encoding Rules (BER) to define the format of encoded (or serialized) messages.

The retrieval and modification operations *set, get, getnext,* and *getbulk* require two messages: a *request* message and a *response* message, as depicted in diagram 80 of FIG. 2A. The first event report operation *trap* uses a single message operation, which is also called *trap*, as shown in diagram 82 of FIG. 2B. The second event report operation *inform* uses two messages: the *inform* message and the *response* message, as illustrated in diagram 84 of FIG. 2C.

An SNMP message includes administrative information and an SNMP Protocol Data UNIT (PDU). The PDU identifies the type of the message. SNMP messages are exchanged between processes called SNMP entities. An SNMP entity may be designed to process only one message at a time (i. e., be implemented as single-threaded), or may be designed to process multiple messages concurrently (i.

e., be implemented as multi-threaded). An SNMP entity can play one of many roles during message exchange. The role is determined by the message type and message originator.

An SNMP entity is called SNMP manager (72 of FIG. 1) if it assumes the operational role to generate requests to retrieve and modify management information, receive the responses to requests, or to receive event reports. An SNMP entity is called SNMP Agent (70 of FIG. 1) if it assumes the operational role to receive, process, and respond to requests, and to generate event reports. An SNMP Agent should have access to network management information to respond to requests, and should be notified of internal events to generate reports. An SNMP entity can also play a dual role of being both an SNMP Agent and an SNMP manager (68 of FIG. 1). However, in describing SNMP message exchange, an entity should perform either as a manager, or as an Agent for a particular message. Dual-role entities are typically used to forward SNMP messages (an SNMP proxy), or to consolidate and synthesize information from many systems and make that information available to a higher-level manager (a mid-level manager). In an SNMP based Client-server model, a Client generally is an SNMP manager, and a server is an SNMP Agent.

For SNMP message exchange, one needs only a simple connectionless (i. e., datagram) transport service. Examples includes the User Datagram Protocol (UDP) from the Internet protocol suite, the Connectionless-mode Transport Service (CLTS) from the OSI protocol suite, the Datagram Delivery Protocol (DDP) from the AppleTalk protocol suite, the Packet Exchange protocol (PEP) from the Novell IPX protocol suite, or any form of interprocess communication message within a single system. The SNMP protocol is stateless, that is the SNMP protocol does not keep information about or for the entities after an operation has been performed. Any such information is kept by the entities themselves.

The second component of the framework is a definition of management information and events that is specified in a plurality of documents called the Structure of Management Information (SMI). The SMI includes the model of management information, the allowed data types, and the rules for specifying classes (or types) of management information.

An SNMP system is managed by retrieval and modification of management information. Each class (or type) of management information is called an *object* or *object type*. A specific instance from a class of management information is called an SNMP *variable* or an *object instance*. The definition of an object type includes its data type, the maximum allowed access, its assigned identity, how instances are identified, and its semantics (or behavior). SNMP uses an identification scheme found in ASN.1 to uniquely identify items for all space and time. An identifier in this scheme is called an *object identifier* (or OID). The permanent assignment of an OID value, an identity, to an item is called *registration*. Once a registration has been performed, no other item may be registered with the same OID value, the characteristics of the registered item may not be changed, and the registered item may not be deleted. SNMP uses OIDs to identify many types of items. Besides the data types and rules for specifying management information, the SMI also includes some administrative assignments. These assignments are used to organize the definitions of management information into those targeted for the Internet Engineering Task Force (IETF) standards track, experimental ones within the IETF, and proprietary ones developed by vendors of SNMP-based management systems. The SMI also includes rules for specifying implementation compliance requirements, and actual implemented characteristics of SNMP Agents.

Definition for related management information, events, and associated implementation compliance requirements are specified in documents called Management Infor-

mation Base (MIB) specifications. These specifications include prose descriptions and computer-readable descriptions. The computer-readable descriptions, called MIB information modules (or just MIB modules), are written in an *adapted* subset of the ASN.1 language.

FIG. 3 illustrates the rules 90 for writing MIB modules that come from two sources: (1) ASN.1 modules 94 defining macros, and defining base type specifications; and (2) textual description 96. The textual description 96 indicate (a) the allowed constructs from ASN.1 language that can be used in MIB modules, (b) the restrictions on those ASN. 1 constructs; and (c) the additions and adaptions to ASN.1 constructs for use in SNMP MIB modules. The rules 90 create a new SNMP MIB language 100 that is used to specify SNMP MIB modules. The SNMP MIB language 100 has common elements 104 with the ASN.1 language 102 that define macros, and additional elements that come from textual descriptions. These additional elements modify the macros and other elements taken from the ASN.1 language. Thus, ASN.1 language is used in SNMP for two different purposes: (1) to define the abstract syntax of SNMP messages; and (2) to specify the format of MIB modules. Computer programs that parse the MIB module language are called MIB compilers.

Referring back to FIG. 1B, in the preferred embodiment the network dual access system 20 of the present invention includes a network_dual_access programmable device 30 and a local_remote_interface device 25 coupled to the network_dual_access programmable device 30. The network_dual_access programmable device 30 can comprise: a router network programmable device; a bridge network programmable device; a network server; or a credit card validation programmable device. On the other hand, the local_remote_interface device 25 can comprise: a laptop computer; or a personal computer.

In the preferred embodiment, the network_dual_access programmable device 30 further comprises a local_remote Simple Network Management Protocol

(SNMP) Agent (L_R_SNMP_Agent) 36 configured to interact with the local_remote_interface device 25 in the dual local_remote access mode utilizing a Simple Network Management Protocol (SNMP). In the preferred embodiment, the local_remote Simple Network Management Protocol (SNMP) Agent (L_R_SNMP_Agent) 36 includes the described above SNMP Agent 70 of FIG. 1C. In the alternative embodiment, the network_dual_access programmable device 30 of FIG. 1B includes a local_remote Common Management Information Protocol (CMIP) Agent (L_R_CMIP_Agent) configured to interact with the local_remote_interface device 25 in the dual local_remote access mode utilizing a prior art Common Management Information Protocol (CMIP).

The SNMP Agent should include the following functional areas: (1) access to one or more transport stacks; (2) a protocol engine which includes security; and (3) a dispatch table to method routines. An SNMP Agent uses a transport stack to receive and transmit messages. An SNMP Agent typically uses a well-known transport selector to receive messages. For example, in the Internet protocol suite, UDP port is used as the transport selector with an IP address of the system including an Agent to form the transport address. The socket interface can be used as an interface between an Agent and a transport stack. The protocol stack should also include instrumentation and access interfaces for method routines at all layers for the management information in the protocol stack to be accessible by the Agent. The protocol engine includes a part of an Agent that receives an SNMP message, decodes it, implements the checks of the administration framework, looks up mappings in the dispatch table, calls method routines, and encodes and sends a response message. The protocol engine also receives notifications of events within the system and generates SNMP event reports. The dispatch table includes mappings to method routines based on management information identity and selector. SNMP messages received or generated by an Agent, include the identity information

(or approximation of the identity of management information in messages for the *getnext* and *getbulk* operations). The identity of an instance of management information includes at least two parts. The first part is the type of management information, and the second part specifies an individual (or instance) of that type. SNMP messages also include a selector which is used to qualify the identities in that message. For example, in SNMPv1, the selector is the community string field in a message, wherein in SNMPv2, the selector is the context field in a message. For each management information identity in a message, an Agent should use the selector from that message, and both parts of the identity to access the management information. An identity and selector should also be specified in requests through a local interface, if this is supported by the Agent. The prior art Agent implementations have tools to automate the generation of a dispatch table, and allows new entries to be created and removed during Agent execution. Adding a new mapping is called *registration*. Deleting a mapping is called *deregistration.*

Referring still to the FIG. 1B, in the preferred embodiment, the network dual access system 20 of the present invention further includes a network port 37 configured to access the network_dual_access programmable device 30 in a remote mode utilizing a Simple Network Management Protocol (SNMP), and a non-network port 23 including a protocol configured to access the network_dual_access programmable device 30 in a local mode using the SNMP protocol. The non-network port 23 can comprise: a serial port, or a parallel port. The serial port can comprise a RS-232 port, or an infrared data access port (IRDA). On the other hand, the parallel port can comprise a general purposes interface bus (GPIB), or a CENTRONIX port. The network port 37 can comprise an Ethernet line, a phone line, a cable modem, or a wireless port configured to wirelessly access the network_dual_access programmable device 30.

Referring still to FIG. 1B, in the preferred embodiment, the local_remote_interface device 25 further includes the local connector 32 and a remote connector (not shown). The local connector 32 is configured to exchange a set of data between the MIB files 24 and the network_dual_access programmable device 30 in the local mode using the SNMP protocol, wherein the remote connector is configured to exchange a set of data between the MIB files 24 and the network_dual_access programmable device 30 in the remote mode using the SNMP protocol and via the Internet 28.

Referring to FIG. 1B, in the preferred embodiment, the local_remote_interface device 25 further includes a management information base (MIB) files 24 including a set of variables configured to be transferred to and/or from the network_dual_access programmable device 30 in the dual local_remote access mode. The MIB files 24 of the local_remote_interface device 25 include variables corresponding to the system parameters 38 of the network_dual_access programmable device 30. In one embodiment, the system parameters are determined during the interrogation session by the SNMP Agent 36 that uses the *get*, and *getnext* commands.

In the preferred embodiment of the present invention, the local_remote_interface device 25 of FIG. 1B includes a configuration management tool 26 configured to read and/or write the set of variables using the management information base (MIB) file 24 in order to configure the graphic user interface (GUI) (not shown). In one embodiment, the configuration management tool 26 can be implemented by using the Open View Interface (OVI) manufactured by Hewlett-Packard (HP), Palo Alto, California.

In the preferred embodiment of the present invention, the ODI configuration tool 26 is used to remotely interrogate the dual access device 30 about its parameters via Internet 28 and UDP port 34 using the SNMP protocol and the SNMP Agent 36 residing on the dual

access programmable device 30. In this embodiment, as was stated above, the system parameters 38 can be also obtained locally by using the SNMP Agent 36. In one embodiment, the SNMP Agent 36 obtains the system parameters 38 in a local mode by using the same interrogation commands *get,* and *getnext.*

EXAMPLES OF *GETNEXT* MODE OF QUESTIONING:

1-st MIB question: What is the programmable device type?
2-nd MIB question: What is the version number?
3-d MIB question: What is the length of time?
4-th MIB question: What is the reset time?

In the preferred embodiment of the present invention, the local_remote_interface device 25 includes a local connector 32 configured to exchange a set of data between the MIB files 24 and system parameters block 38. The system parameters obtained from the block 38 are sent to the MIB Interpreter 21 via command queue pipeline 40. In the preferred embodiment, the command queue pipeline 40 can be implemented by using a common queue 40 to run the connection between the service PC 22 and the SNMP Agent 36 residing on the programmable device 30. The SNMP protocol handles one message at a time, and the common queue pipeline 40 is used to build up the message in both directions. Thus, without any prior knowledge about the system parameters 38, the user can interrogate the programmable device 30 by using SNMP Agent 36. This is possible, because the MIB files topologically are trees of variables, and the topological trunk of all trees is the same for all applications.

In the preferred embodiment, the dual access network system 20 of FIG. 1B utilizes the double access to the programmable device 30 from the local_remote_interface device 25 both locally (via local connector 32, and without trouble of re-writing the program for SNMP), and remotely (via remote connector and

via Internet 28) in order to activate the SNMP Agent 36 that is used to interrogate the programmable device 30 and to find out the updated system parameters 28.

In one embodiment of the present invention, the local_remote_interface device 25 represents a client, and the network_dual_access programmable device 30 represents a server (the so called client-server model). In the SNMP client-server model, as was mentioned above, the SNMP Client resides on the client side, and the SNMP Agent 36 resides on the server side. Therefore, in the SNMP embodiment, the local_remote_interface device 25 further includes a Simple Network Management Protocol (SNMP) Client (not shown), and the network_dual_access programmable device 30 includes the SNMP Agent 36. The SNMP Client includes the *set/get emulator* (not shown) configured to access the network_dual_access programmable device 30 in the dual local/remote access mode utilizing the SNMP protocol.

In one embodiment, the network_dual_access programmable device 30 further includes a data conversion management module (not shown) configured to convert a set of data transferred from and/or to the MIB file 24 using the SNMP protocol, and a data storage system 39 configured to store a set of system parameters 38 that are related to a set of variables stored in the MIB file 24. In the preferred embodiment, the data conversion management module further includes an SNMP interpreter (not shown) configured to interpret a set of data in ASN.1 format transferred to and /or from the MIB file 24.

In one embodiment of the present invention, the local_remote_interface device 25 includes the MIB interpreter configured to interpret the system parameters 38 and to send the updated data to the MIB files 24. The user changes MIB files 24 in his service PC 22 if he needs to update the system parameters 38 of the double access programmable device 30. Thus, to update the system parameters 38, the user uses the same SNMP Agent 36 and the same SNMP protocol in both modes, locally and

remotely. For instance, the technician can service the programmable device 30 locally without writing the new service program for each new feature or application because the new system parameters that correspond to the new features or applications of the device 30 are downloaded to the service PC 22 from the MIB interpreter 21.

In one embodiment, the system parameters 38 include {device type; version number}. In this embodiment, the MIB files 24 set of variables also include {device type; version number}. In another embodiment, the system parameters block 38 include some additional parameters {length_of_time; reset_time}. Thus, for example, the device 30 has to be turned off for the duration of time equal to the parameter {length_of_time}, and has to be reset after the time equal to the parameter {reset_time} lapsed. In this embodiment, the MIB files 24 also include the variables {length_of_time; reset_time}. Yet, in one more embodiment, the system parameters 38 include the enterprise dependent parameters. For example, in a sprinkler network management system, the enterprise dependent variables are:{the number of sprinklers N that should be turned off or on; the id of each particular sprinkler I1, I2, ...IK that has to be turned off or on}, wherein N and K are integers. In this embodiment, the MIB files also include the set variables {the number of sprinklers N that should be turned off or on; the id of each particular sprinkler I1, I2, ...IK that has to be turned off or on}.

FIG. 4 is a flow chart 110 illustrating the method of the present invention describing a dual local_remote SNMP access of a network_dual_access programmable device (30 of FIG. 1B) by using a local_remote_interface device (25 of FIG. 1B). The method of FIG. 4 is performed by the entire dual access network system (20 of FIG. 1B).

At first (step 112 of FIG. 4), an Abstract Syntax Notation One (ASN.1) format derived from a management information based (MIB) file including a set of

variables is transferred from the network_dual_access programmable device (30 of FIG. 1B) to the local_remote_interface device (25 of FIG. 1B). This step is necessary because, as was stated above, the SNMP *operations* occur through *message exchange* over a message transport service.

Next (step 114 of FIG. 4), an SNMP transaction is emulated on the local_remote_interface device (25 of FIG. 1B) by utilizing the MIB file (24 of FIG. 1B) and an SNMP Client (not shown). Indeed, as was explained above, the local_remote_interface device (25 of FIG. 1B) is treated as a client in the SNMP client-server model, and therefore includes an SNMP Client, wherein the network_dual_access programmable device (30 of FIG. 1B) is treated as a server in the SNMP client server model, and accordingly, includes an SNMP Agent (36 of FIG. 1B). The emulated SNMP transaction is transmitted (step 116 of FIG. 4) from the local_remote_interface device to a data conversion module (not shown) installed on the network_dual_access programmable device. Finally (step 118 of FIG. 4), the emulated SNMP transaction is converted into a set of data readable by an SNMP Agent (36 of FIG. 1B).

In the preferred embodiment, the step of transferring the management information base (MIB) file (step 112) further includes the following steps: (step 120 of FIG. 4) reading and/or writing a set of variables in the MIB file (24 of FIG. 1B) by utilizing a configuration management tool (26 of FIG. 1B); and (step 122 of FIG. 4) utilizing the set of variables in the MIB file (24 of FIG. 1B) in order to change configuration of a graphic user interface (GUI) (not shown). In one embodiment, the step of transmitting the emulated SNMP transaction from the local_remote_interface device to the network_dual_access programmable device further includes the step (not shown) of using a local connector (32 of FIG. 1B) in order to make a local connection and to bypass the Internet, and the step (not shown) of using

a remote connector (not shown) in order to make a remote connection via the Internet (28 of FIG.1 B).

In one embodiment, the step of converting the emulated SNMP transaction (step 118 of FIG. 4) further includes ths step of interpreting the emulated SNMP transaction as set of data by utilizing an SNMP interpreter (step 122 of FIG. 4), and the step of reading (step 124) the set of converted data by a local_remote Simple Network Management Protocol (SNMP) Agent (L_R_SNMP_Agent) installed on the network_dual_access programmable device (30 of FIG. 1B). In the alternative embodiment, the step of reading the set of converted data (step 126) is performed by a local_remote Common Management Information Protocol (CMIP) (L_R_CMIP_Agent) Agent (not shown) installed on the network_dual_access programmable device (30 of FIG. 1B).

FIG. 5 illustrates a flow chart 140 of a method of the present invention for a dual local_remote SNMP access of a network_dual_access programmable device. The method 140 of FIG. 5 is performed by the SNMP Client side of the SNMP client-server model. This means that the steps of flowchart 140 are performed by the local_remote_interface device (25 of FIG. 1B).

At first (step 142 of FIG. 5), an Abstract Syntax Notation One (ASN.1) format derived from a management information based (MIB) file including a set of variables is transferred from the network_dual_access programmable device (30 of FIG. 1B) to the local_remote_interface device (25 of FIG. 1B). Next (step 144 of FIG. 5), an SNMP transaction is emulated on the local_remote_interface device (25 of FIG. 1B) by utilizing the MIB file (24 of FIG. 1B) and an SNMP Client (not shown). Finally, the emulated SNMP transaction is transmitted (step 146 of FIG. 5) from the local_remote_interface device to a data conversion module (not shown) installed on the network_dual_access programmable device.

In the preferred embodiment, the step of transferring the management information base (MIB) file (step 142) further includes the following steps: (step 148 of FIG. 5) reading and/or writing a set of variables in the MIB file (24 of FIG. 1B) by utilizing a configuration management tool (26 of FIG. 1B); and (step 150 of FIG.) utilizing the set of variables in the MIB file (24 of FIG. 1B) in order to change configuration of a graphic user interface (GUI) (not shown). In one embodiment, the step of transmitting the emulated SNMP transaction from the local_remote_interface device to the network_dual_access programmable device further includes the step (not shown) of using a local connector (32 of FIG. 1B) in order to make a local connection and to bypass the Internet, and the step (not shown) of using a remote connector (not shown) in order to make a remote connection via the Internet (28 of FIG.1 B).

The description of the preferred embodiment of this invention is given for purposes of explaining the principles thereof, and is not to be considered as limiting or restricting the invention since many modifications may be made by the exercise of skill in the art without departing from the scope of the invention.

Claims

1. A network dual access system comprising:

a network_dual_access programmable device configured to be accessed using a dual local_remote mode; and

a local_remote_interface device coupled to said network_dual_access programmable device;

wherein said network_dual_access programmable device utilizes said local_remote_interface device for information transfer in said dual local_remote access mode.

2. The system of claim 1, wherein said network_dual_access programmable device further comprises:

a local_remote Network Management Protocol (NMP) Agent (L_R_NMP_Agent) configured to interact with said local_remote_interface device in said dual local_remote access mode utilizing a Network Management Protocol (NMP).

3. The system of claim 1, wherein said network_dual_access programmable device further comprises:

a local_remote Simple Network Management Protocol (SNMP) Agent (L_R_SNMP_Agent) configured to interact with said local_remote_interface device in said dual local_remote access mode utilizing a Simple Network Management Protocol (SNMP).

4. The system of claim 1, wherein said network_dual_access programmable device further comprises:

a local_remote Common Management Information Protocol (CMIP) Agent (L_R_CMIP_Agent) configured to interact with said local_remote_interface programmable device in said dual local_remote access mode utilizing a Common Management Information Protocol (CMIP).

5. The system of claim 3, wherein said local_remote_interface device further includes:

a management information base (MIB) file including a set of variables configured to be transferred to and/or from said network_dual_access programmable device in said dual local_remote access mode.

6. The system of claim 5, wherein said local_remote_interface device further comprises:

a configuration management tool configured to read and/or write said set of variables using said management information base (MIB) file in order to configure the graphic user interface (GUI).

7. The system of claim 6, wherein said local_remote_interface device further comprises:

a local connector configured to exchange a set of data between said MIB and said network_dual_access programmable device in said local mode using said SNMP protocol and bypassing the Internet.

8. The system of claim 6, wherein said local_remote_interface device further comprises:

a remote connector configured to exchange a set of data between said MIB and said network_dual_access programmable device in said remote mode using said SNMP protocol and via the Internet.

9. The system of claim 8, wherein said local_remote_interface device further includes:

a Simple Network Management Protocol (SNMP) Client configured to access said network_dual_access programmable device in said dual local_remote access mode utilizing said Simple Network Management Protocol (SNMP).

10. The system of claim 1, wherein said network_dual_access programmable device further includes:

a data conversion management module configured to convert a set of data transferred from and/or to said MIB file using said SNMP protocol .

11. The system of claim 10, wherein said data conversion management module further includes:

an SNMP interpreter configured to interpret a set of data in ASN.1 format transferred to and /or from said MIB file.

12. The system of claim 1, wherein said network_dual_access programmable device further includes:

a data storage system configured to store a set of network_dual_access programmable device-based parameters that are related to a set of local_remote_interface programmable device-based parameters stored in said MIB file.

13. A local_remote SNMP access system configured to access a network_dual_access programmable device comprising:

a network port configured to access said network_dual_access programmable device in a remote mode utilizing a Simple Network Management Protocol (SNMP); and

a non-network port configured to access said network_dual_access programmable device in a local mode using said SNMP protocol.

14. The system of claim 13, wherein said local_remote SNMP access system further includes:

a management information base (MIB) file including a set of variables configured to be transferred to and/or from said network_dual_access programmable device in said dual local_remote access mode.

15. The system of claim 14, wherein said local_remote SNMP access system further comprises:

a configuration management tool configured to read and/or write said set of variables using said management information base (MIB) file in order to configure the graphic user interface (GUI).

16. The system of claim 15, wherein said local_remote SNMP access system further comprises:

a local connector configured to exchange a set of data between said MIB and said network_dual_access programmable device in said local mode using said NMP protocol and bypassing the Internet.

17. The system of claim 16, wherein said local_remote SNMP access system further comprises:

a remote connector configured to exchange a set of data between said MIB and said network_dual_access programmable device in said remote mode using said SNMP protocol and via the Internet.

18. The system of claim 17, wherein said local_remote SNMP access system further includes:

a Simple Network Management Protocol (SNMP) Client configured to access said network_dual_access programmable device in said dual local_remote access mode utilizing said Simple Network Management Protocol (SNMP).

19. The system of claim 13, wherein said non-network port further comprises:

a serial port.

20. The system of claim 19, wherein said serial port further comprises:

a RS-232 port.

21. The system of claim 19, wherein said serial port further comprises:

an infrared data access port (IRDA).

22. The system of claim 13, wherein said non-network port further comprises:

a parallel port.

23. The system of claim 22, wherein said parallel port further comprises:

a general purposes interface bus (GPIB).

24. The system of claim 22, wherein said parallel port further comprises:

a CENTRONIX port.

25. The system of claim 13, wherein said network port further comprises:

an Ethernet line configured to access said network_dual_access programmable device.

26. The system of claim 13, wherein said network port further comprises:

a phone line configured to access said network_dual_access programmable device.

27. The system of claim 13, wherein said network port further comprises:

a cable modem configured to access said network_dual_access programmable device.

28. The system of claim 13, wherein said network port further comprises:

a wireless port configured to wirelessly access said network_dual_access programmable device.

29. The system of claim 1, wherein said network_dual_access programmable device further comprises:

a router network programmable device.

30. The system of claim 1, wherein said network_dual_access programmable device further comprises:

a bridge network programmable device.

31. The system of claim 1, wherein said network_dual_access programmable device further comprises:

a network server.

32. The system of claim 1, wherein said network_dual_access programmable device further comprises:

a credit card validation programmable device.

33. The system of claim 1, wherein said local_remote_interface device further comprises:

a laptop computer.

34. The system of claim 1, wherein said local_remote_interface device further comprises:

a personal computer.

35. A method for a dual local_remote SNMP access of a network_dual_access programmable device using a local_remote_interface device, said method comprising the steps of:

transferring an Abstract Syntax Notation One (ASN.1) format derived from a management information based (MIB) file including a set of variables from said network_dual_access programmable device to said local_remote_interface device;

emulating an SNMP transaction by utilizing said MIB file and by using a Simple Network Management Protocol (SNMP) Client installed on said local_remote_interface device;

transmitting said emulated SNMP transaction from said local_remote_interface device to a data conversion module installed on said network_dual_access programmable device;

and

converting said emulated SNMP transaction into a set of data readable by an Agent installed on said network_dual_access programmable device.

36. The method of claim 35, wherein said step of transferring said management information base (MIB) file further includes the steps of:

reading and/or writing a set of variables in said MIB file by utilizing a configuration management tool; and

utilizing said set of variables in said MIB file in order to change configuration of a graphic user interface (GUI).

37. The method of claim 35, wherein said step of transmitting said emulated SNMP transaction from said local_remote_interface device to said network_dual_access programmable device further includes the step of:

using a local connector installed on said local_remote_interface device in order to make a local connection and to bypass the Internet.

38. The method of claim 35, wherein said step of transmitting said emulated SNMP transaction from said local_remote_interface device to said network_dual_access programmable device further includes the step of:

using a remote connector installed on said local_remote_interface device in order to make a remote connection via the Internet.

39. The method of claim 35, wherein said step of converting said emulated SNMP transaction further includes ths step of:

interpreting said emulated SNMP transaction as set of data by utilizing an SNMP interpreter.

40. The method of claim 35, wherein said step of converting said emulated SNMP transaction further includes ths step of:

reading said set of converted data by a local_remote Simple Network Management Protocol (SNMP) Agent (L_R_SNMP_Agent)installed on said network_dual_access programmable device.

41. The method of claim 35, wherein said step of converting said emulated SNMP transaction further includes ths step of:

reading said set of converted data by a local_remote Common Management Information Protocol (CMIP) (L_R_CMIP_Agent) Agent installed on said network_dual_access programmable device.

42. A method for a dual local_remote SNMP access of a network_dual_access programmable device using a local_remote_interface device, said method comprising the steps of:

transferring an ASN. 1 format derived from an MIB file including a set of variables from said network_dual_access programmable device to said local_remote_interface device;

emulating an SNMP transaction by utilizing said MIB file and by using a Simple Network Management Protocol (SNMP) Client installed on said local_remote_interface device;

and

transmitting said emulated SNMP transaction from said local_remote_interface device to a data conversion module installed on said network_dual_access programmable device.

43. The method of claim 42, wherein said step of transferring said management information base (MIB) file further includes the steps of:

reading and/or writing a set of variables in said MIB file by utilizing a configuration management tool; and

utilizing said set of variables in said MIB file in order to change configuration of a graphic user interface (GUI).

44. The method of claim 42, wherein said step of transmitting said emulated SNMP transaction from said local_remote_interface device to said network_dual_access programmable device further includes the step of:

using a local connector installed on said local_remote_interface device in order to make a local connection and to bypass the Internet.

45. The method of claim 42, wherein said step of transmitting said emulated SNMP transaction from said local_remote_interface device to said network_dual_access programmable device further includes the step of:

using a remote connector installed on said local_remote_interface device in order to make a remote connection via the Internet.

46. A system for a dual local_remote SNMP access of a network_dual_access programmable device comprising:

a means for transferring an ASN.1 format derived from an MIB file including a set of variables from said network_dual_access programmable device;

a means for emulating an SNMP transaction by utilizing said MIB file;

a means for locally or remotely transmitting said emulated SNMP transaction to said network_dual_access programmable device;

and

a means for converting said emulated SNMP transaction into a set of data readable by an SNMP Agent installed on said network_dual_access programmable device.

47. A system for a dual local_remote SNMP access of a network_dual_access programmable device comprising:

a means for transferring an ASN.1 format derived from an MIB file including a set of variables from said network_dual_access programmable device;

a means for emulating an SNMP transaction by utilizing said MIB file;

and

a means for locally or remotely transmitting said emulated SNMP transaction to said network_dual_access programmable device.

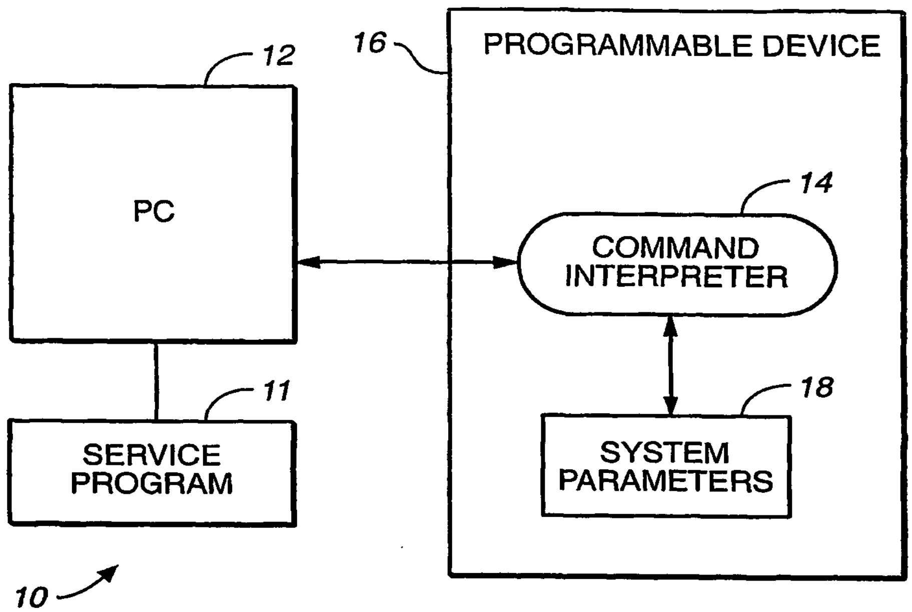
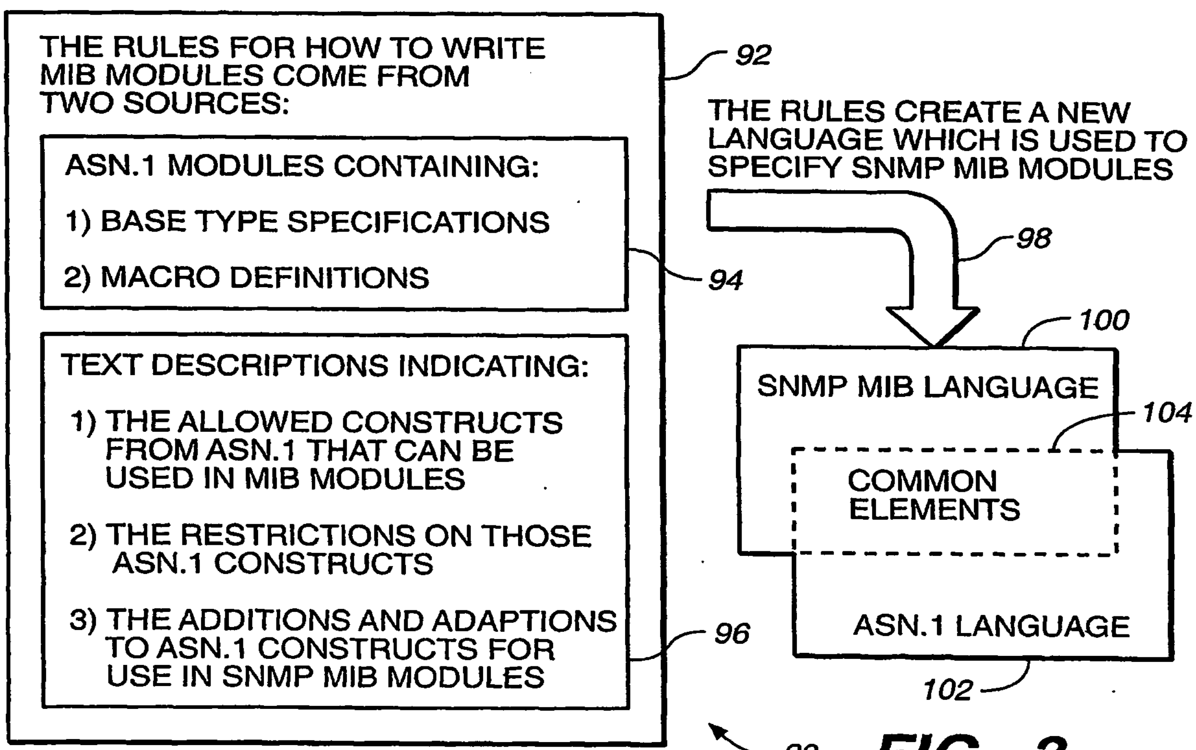


FIG._1A



1 RULE FOR WRITING MIB MODULES

FIG._3
(PRIOR ART)

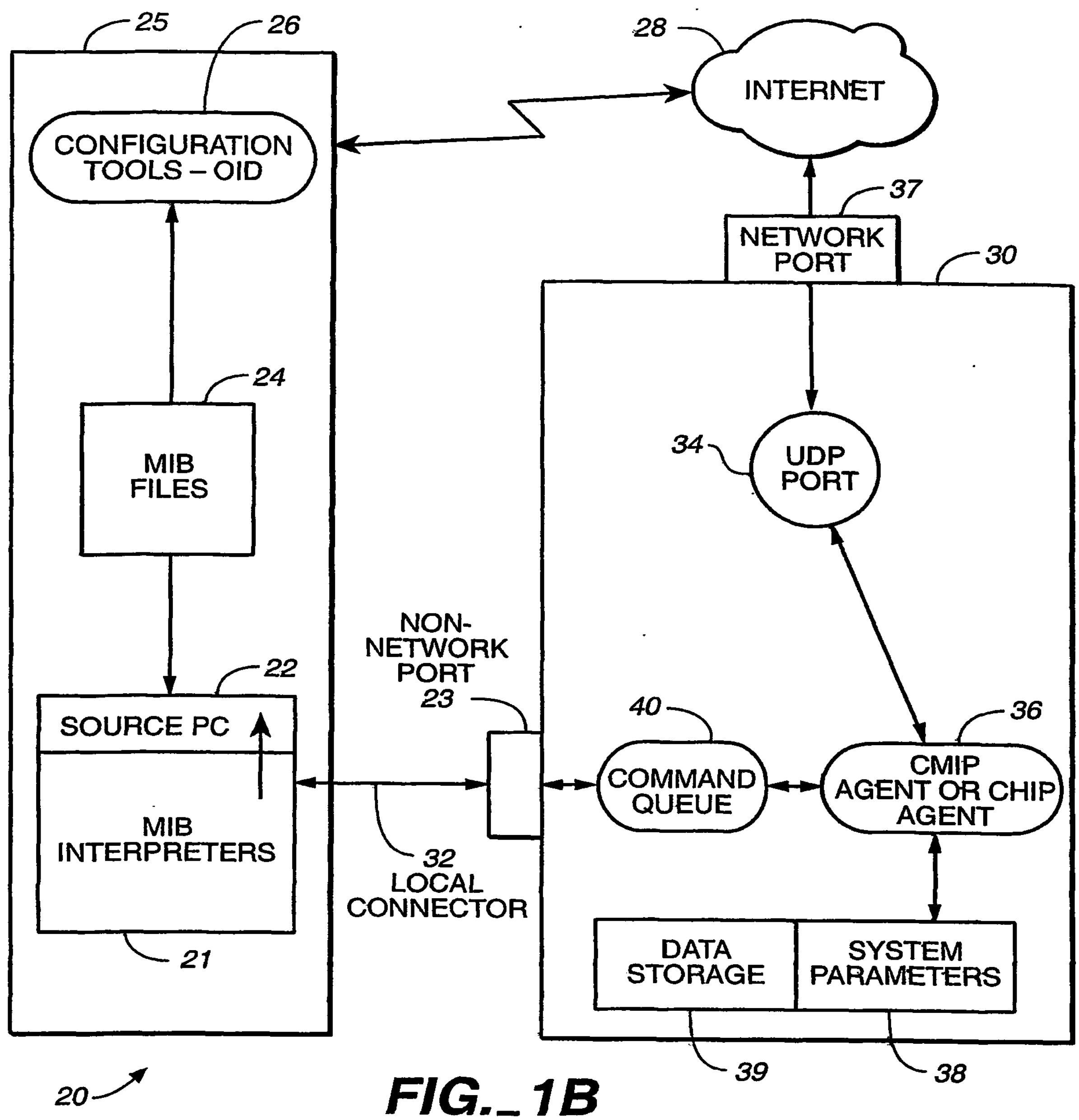
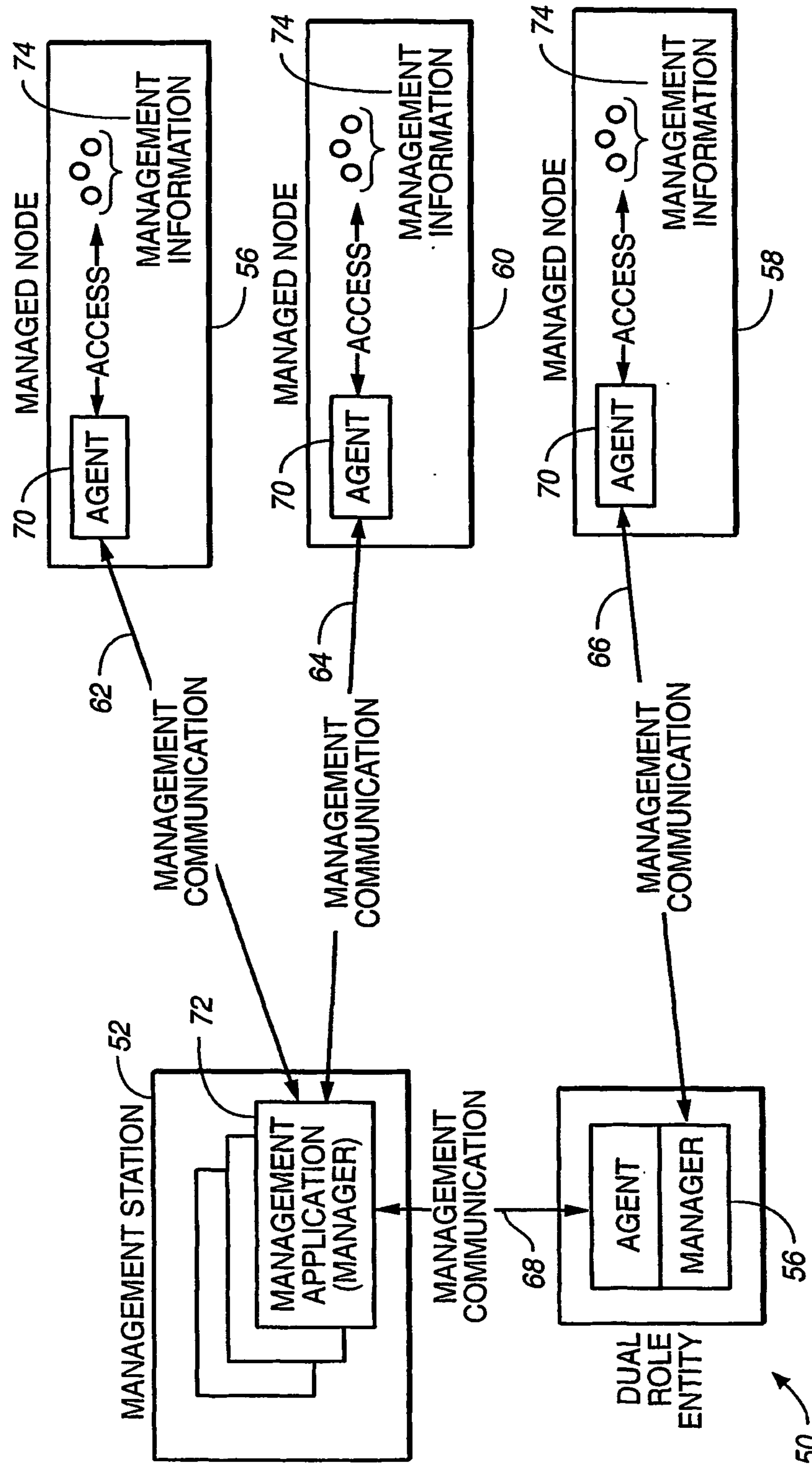

25
26
28
INTERNET
CONFIGURATION TOOLS – OID
37
NETWORK PORT
30
24
MIB FILES
34
UDP PORT
NON-NETWORK PORT
23
22
SOURCE PC
MIB INTERPRETERS
40
COMMAND QUEUE
36
CMIP AGENT OR CHIP AGENT
32
LOCAL CONNECTOR
21
DATA STORAGE
SYSTEM PARAMETERS
39
38
20

FIG._1B



THE SNMP MODEL OF A MANAGED NETWORK

FIG._1C
(PRIOR ART)

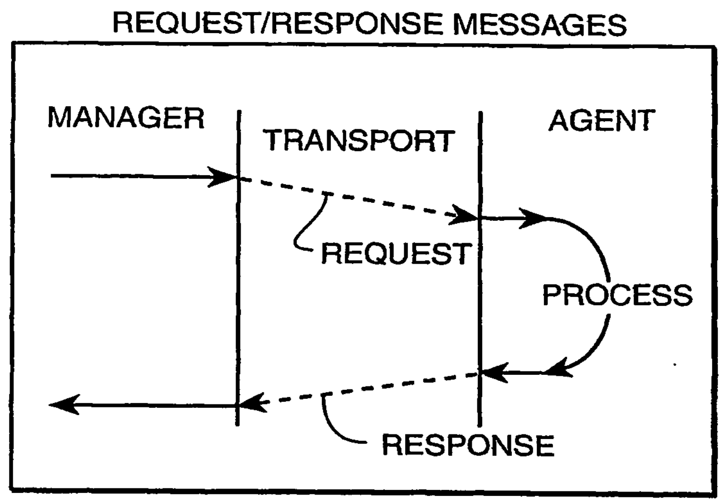


FIG._2A
(PRIOR ART)

80

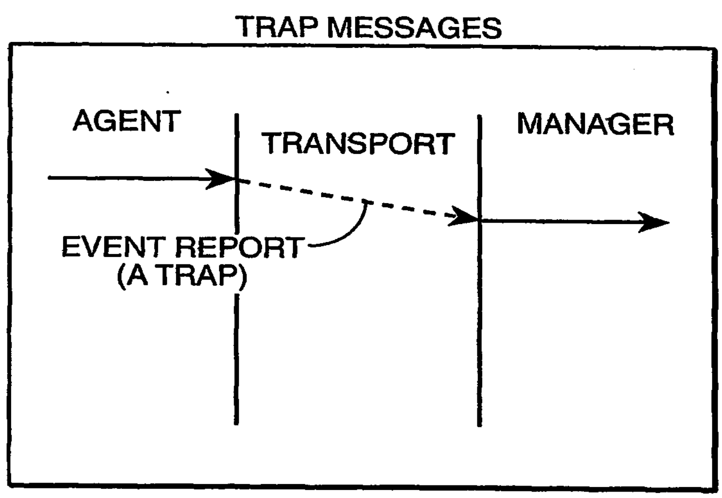


FIG._2B
(PRIOR ART)

82

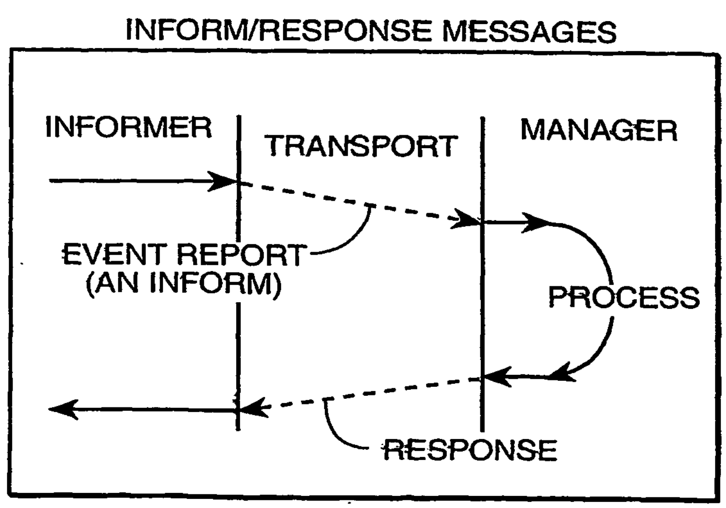


FIG._2C
(PRIOR ART)

84

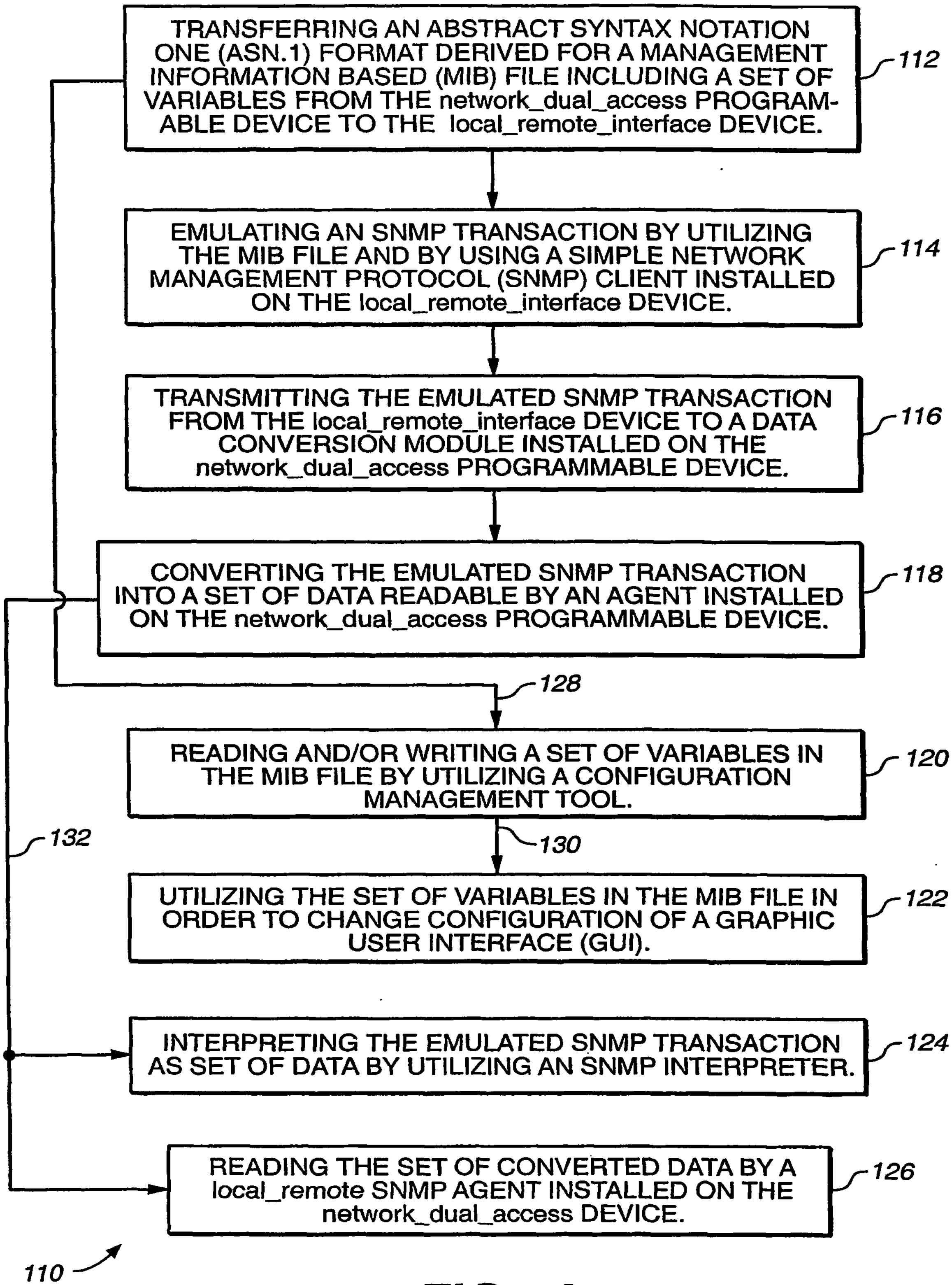

TRANSFERRING AN ABSTRACT SYNTAX NOTATION ONE (ASN.1) FORMAT DERIVED FOR A MANAGEMENT INFORMATION BASED (MIB) FILE INCLUDING A SET OF VARIABLES FROM THE network_dual_access PROGRAMMABLE DEVICE TO THE local_remote_interface DEVICE.
112
EMULATING AN SNMP TRANSACTION BY UTILIZING THE MIB FILE AND BY USING A SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP) CLIENT INSTALLED ON THE local_remote_interface DEVICE.
114
TRANSMITTING THE EMULATED SNMP TRANSACTION FROM THE local_remote_interface DEVICE TO A DATA CONVERSION MODULE INSTALLED ON THE network_dual_access PROGRAMMABLE DEVICE.
116
CONVERTING THE EMULATED SNMP TRANSACTION INTO A SET OF DATA READABLE BY AN AGENT INSTALLED ON THE network_dual_access PROGRAMMABLE DEVICE.
118
128
READING AND/OR WRITING A SET OF VARIABLES IN THE MIB FILE BY UTILIZING A CONFIGURATION MANAGEMENT TOOL.
120
130
132
UTILIZING THE SET OF VARIABLES IN THE MIB FILE IN ORDER TO CHANGE CONFIGURATION OF A GRAPHIC USER INTERFACE (GUI).
122
INTERPRETING THE EMULATED SNMP TRANSACTION AS SET OF DATA BY UTILIZING AN SNMP INTERPRETER.
124
READING THE SET OF CONVERTED DATA BY A local_remote SNMP AGENT INSTALLED ON THE network_dual_access DEVICE.
126
110

FIG._4

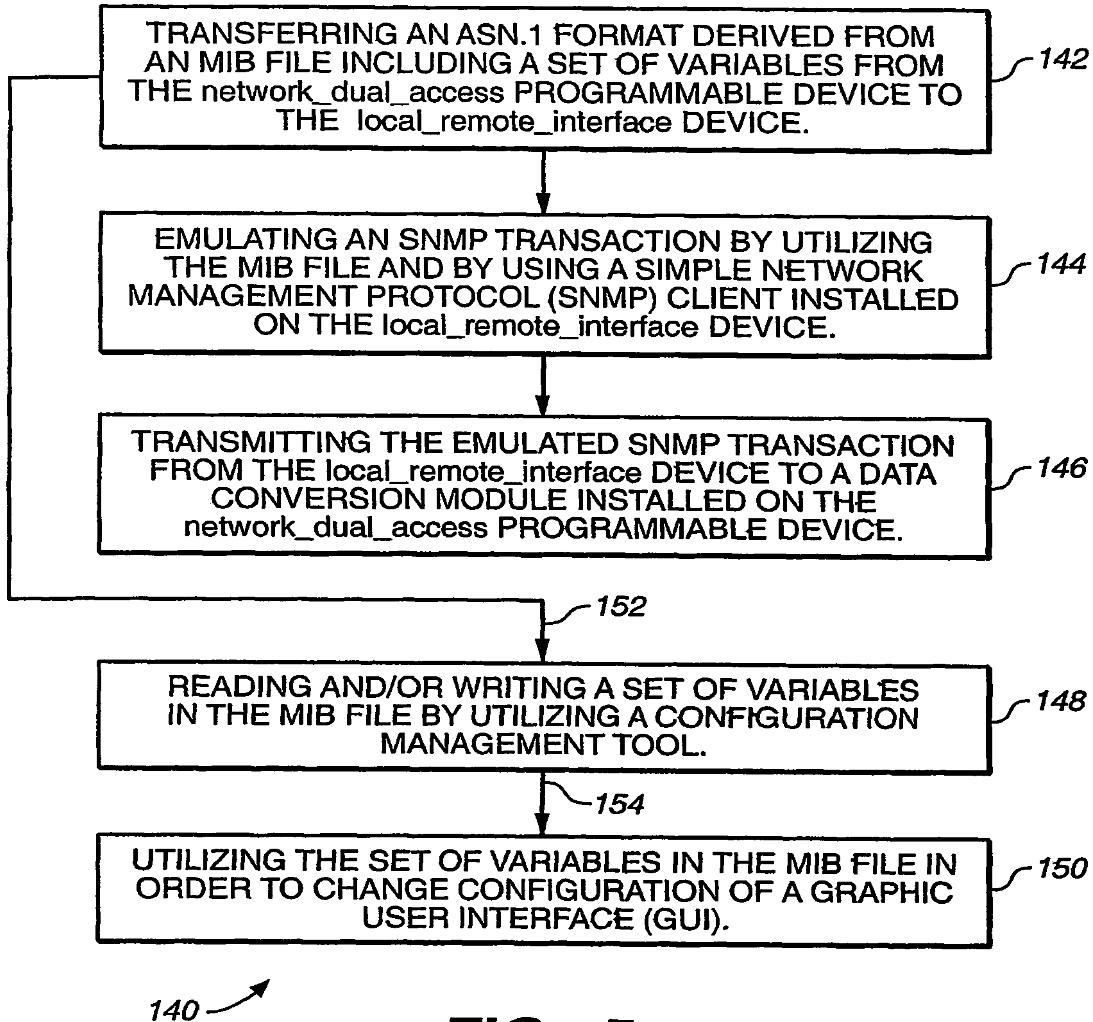

TRANSFERRING AN ASN.1 FORMAT DERIVED FROM AN MIB FILE INCLUDING A SET OF VARIABLES FROM THE network_dual_access PROGRAMMABLE DEVICE TO THE local_remote_interface DEVICE.
142
EMULATING AN SNMP TRANSACTION BY UTILIZING THE MIB FILE AND BY USING A SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP) CLIENT INSTALLED ON THE local_remote_interface DEVICE.
144
TRANSMITTING THE EMULATED SNMP TRANSACTION FROM THE local_remote_interface DEVICE TO A DATA CONVERSION MODULE INSTALLED ON THE network_dual_access PROGRAMMABLE DEVICE.
146
152
READING AND/OR WRITING A SET OF VARIABLES IN THE MIB FILE BY UTILIZING A CONFIGURATION MANAGEMENT TOOL.
148
154
UTILIZING THE SET OF VARIABLES IN THE MIB FILE IN ORDER TO CHANGE CONFIGURATION OF A GRAPHIC USER INTERFACE (GUI).
150
140

FIG._5

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/09780

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/173, 15/16
US CL :709/223, 225, 250; 370/444
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 709/223, 225, 250; 370/444

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EAST

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,757,801 A (ARIMILLI) 26 MAY 1998, figures 4B and 6B, col. 10, lines 7-14.	1 and 29-34
A	US 5,634,009 A (IDDON ET AL.) 27 MAY 1997, col. 1, line 15 - col. 4, line 41, col. 5, line 43 - col. 6, line 61.	2-28 and 35-47

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
18 MAY 2001	27 JUN 2001

Name and mailing address of the ISA/US	Authorized officer
Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Peggy Harrod LE HIEN LUU
Facsimile No. (703) 305-3230	Telephone No. (703) 305-9650